TUMBLEWEED COMMUNICATIONS

Digital Certificate Validation in Public Key Infrastructures (PKI), and the Online Certificate Status Protocol (OCSP)

A TUMBLEWEED WHITE PAPER

Digital certificates provide individuals and companies with virtual IDs, or credentials, for use in the electronic world. However, like passports or credit cards, these digital credentials can expire, be revoked, or be suspended. Enterprises that conduct business electronically need to confirm that a digital certificate is current and valid before allowing a transaction to occur, much in the same way that a merchant needs to check a credit card before processing a purchase. The Tumbleweed Valicert Validation Authority (VA) delivers a comprehensive, scalable, and reliable framework for real time validation of digital certificates issued by any certificate authority (CA).

Overview

Many enterprises have deployed a Public Key Infrastructure (PKI) in order to use digital certificates to address their organization's security needs. While digital certificates are an excellent way to help establish the identities of parties wishing to communicate securely or engage in electronic transactions, like any other credentials, digital certificates can be trusted only if they are shown to be valid at the time they are presented. Digital certificates alone do not ensure security.

The world abounds with examples where trust in a credential is achieved only after its validity is established. Drivers' licenses and passports are obvious examples of credentials that are routinely verified when presented. In fact, the most analogous case is that of the ubiquitous credit card because it, too, is a vehicle for trust in business transactions. No merchant would think of accepting a credit card for payment without first receiving authorization for the purchase from a central validating authority. After all, the card could be stolen or counterfeited, or the customer could be over their credit limit. Verifying the validity of the credit card is part of the legal framework for their use, protecting the merchant from fraud and limiting liability. Similarly people constantly leave organizations, consultants come and go, laptops get stolen, and people's statuses change. Their certificates have to reflect those changes. Unless applications are validating certificates, there is no way to know if they can trust them.

This entails certain legal ramifications as well. If a relying party does not know whether the digital certificates they are relying upon are valid or not, then they do not know if the contract that is in place, the binding agreement called the practices statement, is valid. Practices statements clearly maintain that the contract is binding only if the certificate is in its operational period, which is defined as not expired and not revoked. A relying party, therefore, must know whether or not a certificate presented to them is revoked in order to know if the contract that backs up the certificate is in place.

Digital certificate validation is the Achilles heel of many of the PKI's now being put in place to protect parties engaging in secure transactions. Without a reliable way of checking the validity of every digital certificate presented to an application, the relying parties in a transaction have no legal recourse should someone use a certificate in some rogue or malicious fashion.

The problem isn't a lack of information about which certificates have been revoked. There is plenty of that to go around. Rather, it is the hodgepodge of incompatible certificate authorities, validation protocols and

directory services, plus the fact that many certificate-based applications don't even perform validity checks that leaves many PKI's vulnerable.

Background

A PKI uses private-public key cryptography to provide entity authentication, non-repudiation, data integrity, data confidentiality, and access control. PKI relies on digital certificates, defined in the ISO X509v3 standard, containing information identifying a user and their public key. A Certificate Authority (CA) issues digital certificates. A CA receives the public key from a user's private-public key pair along with some information regarding the user's identity. When the CA has verified the supplied data to its satisfaction, it issues a digital certificate signed with its own private key. At this point anyone who receives a certificate and trusts the CA can trust that the user who supplied the certificate is who they say they are, and that the public key contained in the certificate belongs to the certificate owner.

Validation

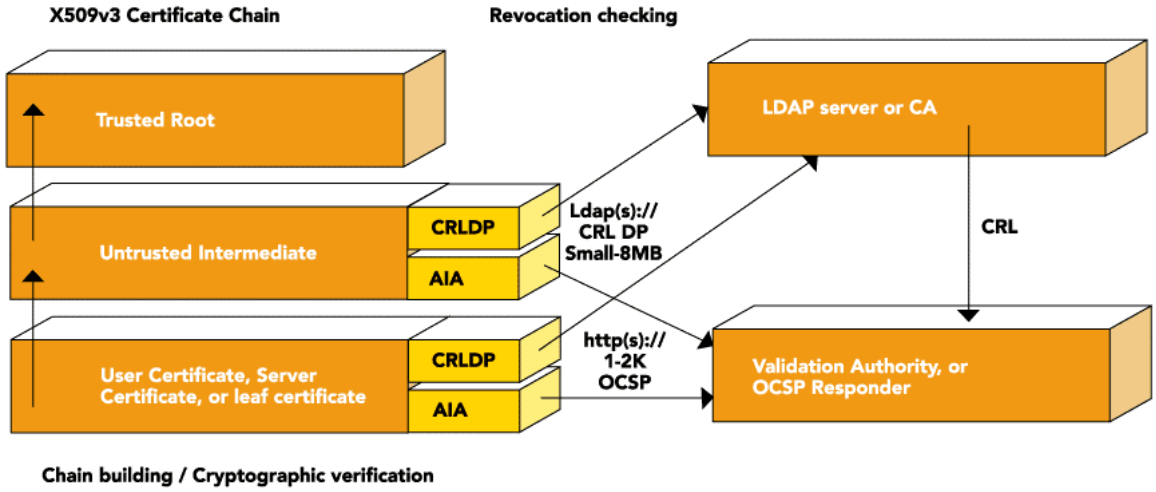

Figure 1:

Figure 1 shows how a trust chain, built within a client or server product is checked cryptographically. Cryptographic checking only ensures that the digital certificates haven't been tampered with, but it doesn't provide any assurances that they are no longer valid. The following terms are useful to keep in mind:

Trusted Root: A trusted root is an X509v3 certificate where the subject name and issuer name are the same. A trusted root represents the end point in the chain building process. Once a trusted root is chain to, the chaining process stops.

Untrusted Intermediate: This certificate isn't considered trusted, so its contents need to be verified. This is done by taking the public key of the issuers certificate (e.g. the trusted root), and making sure that if the untrusted intermediate certificate contents is hashed and this hash decrypted with the public key from the issuers certificate. If the decrypted hash matches the hash of the certificate, then the certificate is assumed to be unaltered.

User Certificate, Server Certificate, or leaf certificate: these certificates usually are from secure servers, secure e-mail clients, or any PKI enabled application. It is usually the leaf certificates that need the most validation checking. The process for untrusted intermediates is the same for cryptographic chain verification.

Chain building: looking for a certificate based on a certificates issuer field ; this process occurs iteratively until a trusted root is encountered.

Cryptographic verification: The process of obtaining a certificate issuers pubic key to verify the hash of a current certificate hash. If the decryption of the signature (hash of certificate) matches a computed hash of the certificate, then it is considered unaltered.

CRL DP: Certificate Revocation List (CRL) Distribution point: certificate extension: This extension points to the location of where a full CRL can be obtained: this could be an ldap://, ldaps://, http:// or https:// URL.

Authority Information Access (AIA) certificate extension: This extension is usually a pointer to an http:// or https:// location where an OCSP responder will be available to provide an OCSP response. OCSP request/response is typically 1-2k in size, vs. a CRL DP that can range in size from many kilobytes to several megabytes.

Since it is possible for digital certificates to become expired, revoked, or suspended, a user cannot be trusted until the status of their digital certificate is validated. Periodically, a CA publishes a Certificate Revocation List (CRL), a complete list of all the digital certificates that have been revoked or suspended. If the user's certificate is not expired and if it does not appear on the CRL, then the digital certificate can be assumed valid, and the user trusted. Unfortunately, due to the cumulative nature of a CA published CRL, requiring each application to obtain and check the complete CA published CRL is a fundamentally unscalable solution. For a client to download a CRL (typically by following the certificates CRLDP), a client may fetch anywhere from a few kilobytes of data to 8 MB of data. Also, since the CA simply publishes a CRL, there is no audit mechanism in terms of digital certificate validation in a certain transaction. Additionally, in most real world PKI deployments, applications do not have real-time network access to the actual CA, which is secured from potential attacks via a private network protected by firewalls. These issues are addressed through the introduction of a Validation Authority (VA) in the PKI.

Validation Authorities (VA)

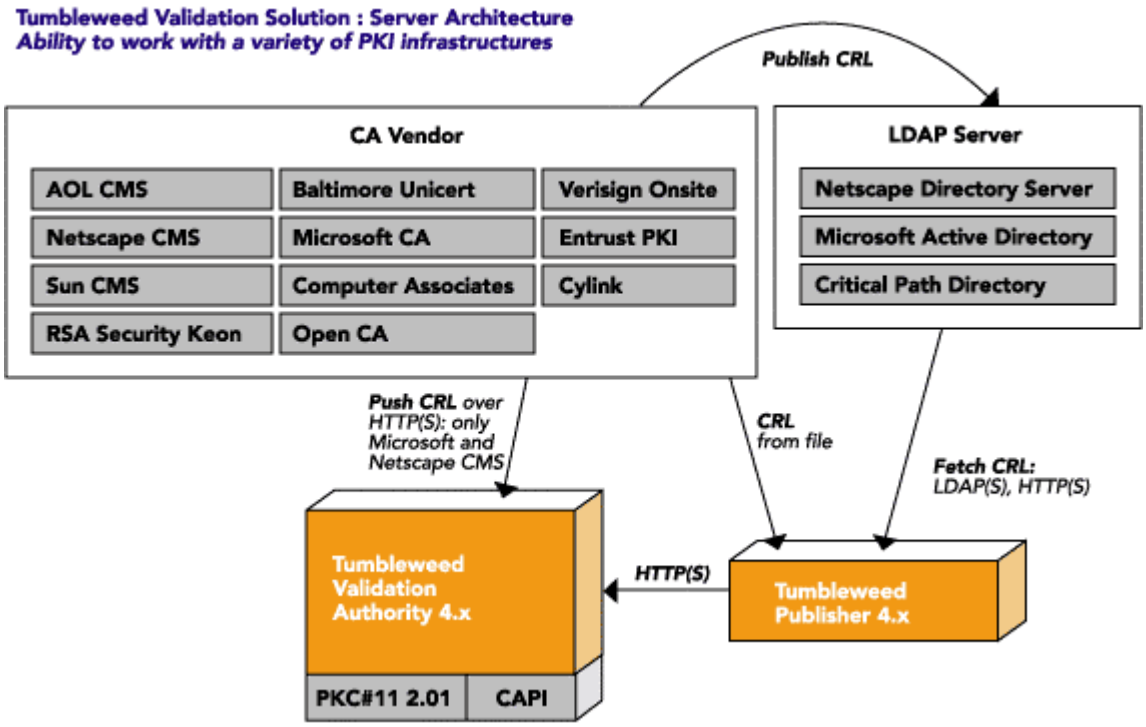

Figure 2:

A Validation Authority (VA) provides a universal clearing house for establishing the validity of a digital certificate. Figure 2 illustrates a typical PKI, which includes a VA. The VA represents a centralized store of aggregated CA published CRL's. It is possible for a VA to aggregate CRL's from one or more different CA's. This store of certificate status data is continuously available and accessible to PKI enabled applications via several standard real-time protocols. These protocols, which will be discussed in greater detail, allow PKI applications to obtain the status of a specific certificate rather than the raw cumulative CRL periodically published by the CA (as illustrated above in Figure 1). Thus the introduction of a VA addresses the scalability and access issues associated with CA certificate validation in PKI, as well as the audit requirements for secure transactions.

Certificate Validation Trust Models

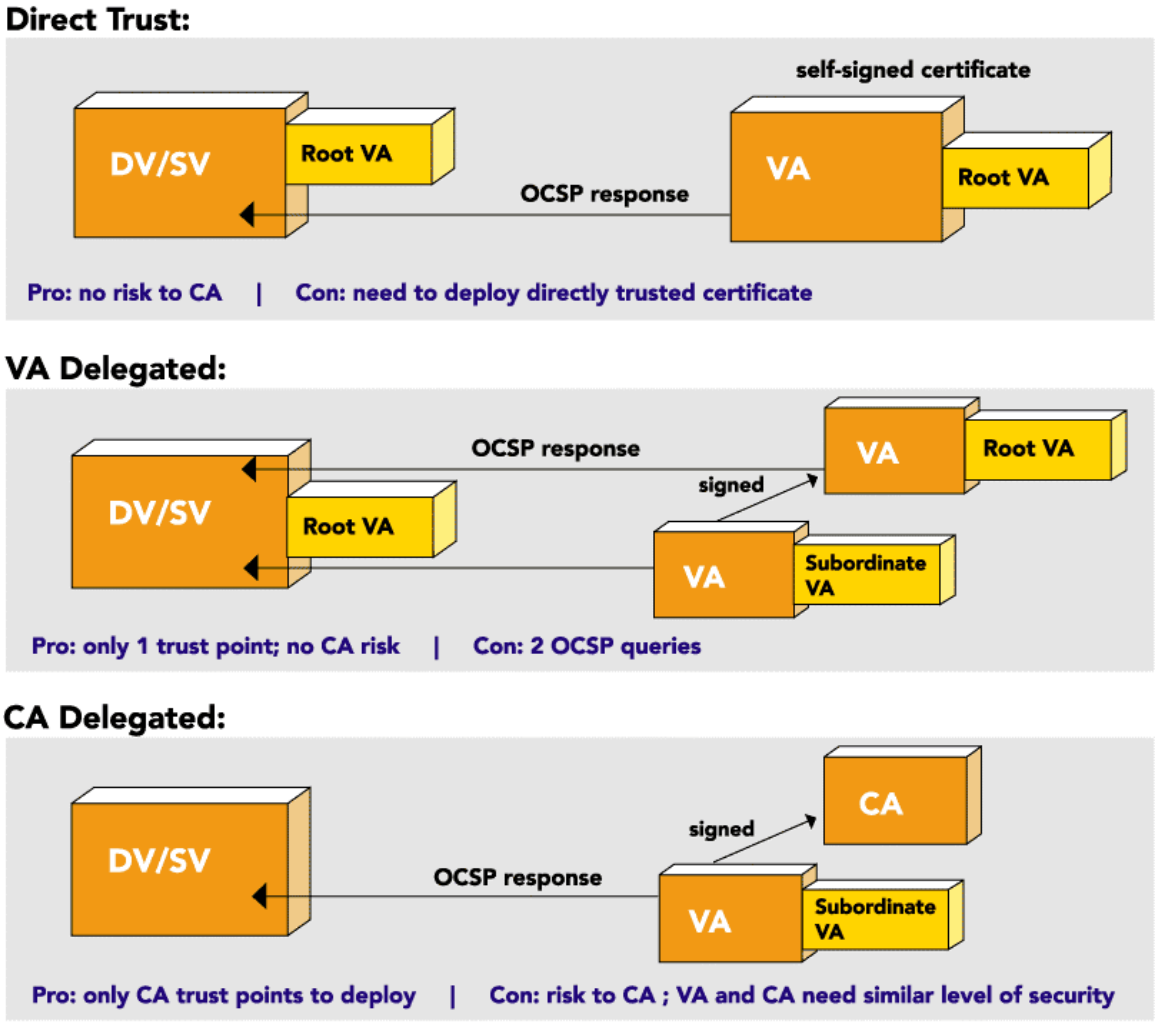

Figure 3:

Certificate validation implies trust between the client and the VA. There are three models for this trust: Direct, VA-delegated, and CA-delegated.

Direct

Direct trust occurs when the client requesting validation has set up a direct trust relationship with the VA. Since the client directly trusts the VA certificate as a trusted authority, the authority can sign revocation responses with a self-issued digital certificate on behalf of any CA. The advantage of this scheme is that it requires no trust chain to be established under a CA, thus reducing the risk/exposure to the CA for creating this chain. The disadvantage of this approach is that the directly trusted VA certificate needs to be distributed to the applications that will be making validation queries. When the VA certificate expires, the clients will need to be updated to trust a new directly trusted VA certificate.

VA Delegated

VA delegated trust is a derivative of the direct trust model which enables a directly trusted VA to delegate its responsibilities to a "subordinate" VA further addressing scalability and operational domain issues that often come up in PKI. The directly trusted VA has a self-issued digital certificate referred to as a root VA certificate. The directly trusted VA uses this root certificate to issue a digital certificate to a subordinate VA, which in turn uses this digital certificate to sign client responses. In order for a client to trust the subordinate VA it must establish the certificate chain back to the directly trusted VA. The certificate depth is limited to one level, meaning the subordinate VA certificates can only be used to sign client responses, and cannot be used to issue other signing certificates.

This model is operationally more secure since a particular subordinate VA could be compromised without compromising the directly trusted VA (root VA) or any other subordinate VA. Additionally, since the VA root certificate is self-issued, the VA can also provide a check and balance for the trustworthiness of the CA itself, ensuring trust in the CA by providing the mechanism for revoking it. If the CA itself is breached, perhaps by someone posing as an employee of the CA and issuing their own bogus certificates, the VA is able to revoke the entire CA if needed. The VA Delegated model has its advantages: only one trust point for the root VA needs to be distributed to clients. However, an extra OCSP query needs to be sent to check the subordinate VA certificate to make sure it has not been revoked. With OCSP caching support at the clients, OCSP responses for common certificates (e.g. subordinate VA certificates) can be held for a configurable amount of time, reducing the number of these queries required.

CA Delegated

CA delegated trust occurs when a CA has explicitly given permission to a VA to respond to revocation requests on its behalf. This is similar to the VA Delegated trust model described above except that the certificate used by the VA to sign responses chains back to the issuing CA rather than to the directly trusted VA. In order to trust the VA, the client must still establish the certificate chain back to the CA, but in many cases this may not require any additional operations since the client already trusts the CA. This model relies on certain extensions to the digital certificates that all participants in the PKI must recognize. Additionally, since the CA and VA are under different administrational boundaries in most operational environments, a CA is potentially opening itself up for liability by delegating validation to a different entity. This model has some clear advantages: only CA trust points need to be distributed. The main disadvantage is that the CA must delegate a CA responsibility to a VA. If the VA is compromised, the CA is compromised, with respect to the integrity of the status information known about the CA. This is why for CA Delegated mode, the CA and VA servers need to have almost equivalent physical security requirements.

Validation Protocols

A VA offers a server referred to as a Responder to handle PKI client application requests for digital certificate status. The client may interact with the Responder in several different ways.

Online Certificate Status Protocol (OCSP)

OCSP, defined by the IETF in RFC2560 (http://www.ietf.org/rfc/rfc2560.txt?number=2560), enables applications to determine whether an identified certificate(s) appears in a CA published CRL by querying a

Responder. The Responder consults its store (built by obtaining CA published CRL's) and returns the status of the certificate(s) identified, including potential revocation information (such as at what time the certificate was revoked and the reason for the revocation). Additionally, for the transaction to be secure and auditable, the Responder will include the time at which the response was produced and the length of time for which this response can be trusted, digitally signing the entire response using its VA certificate.

Since the Responder does not have access to the actual certificate being checked, prior to submitting its request to the Responder, the client must:

- Cryptographically verify that the subject certificate was in fact issued by the corresponding CA
- Verify that the issuing CA is indeed trusted for signing certificates.
- Verify that the subject and issuer certificates are time valid i.e., not expired

Once it receives the Responder's answer, the client must establish whether it trusts the VA as per the trust models previously discussed. OCSP is conducted over HTTP or HTTP/SSL and TCP/IP and benefits from the ubiquity of these underlying protocols.

Simple Certificate Validation Protocol (SCVP)

SCVP is an IETF defined protocol still in draft format (http://www.ietf.org/internet-drafts/draft-ietf-pkix-scvp-13.txt) that would allow a client to completely offload certificate handling to the Responder. Rather than sending just certificate identification information as in the OCSP protocol, a client using the SCVP protocol would send the entire certificate to the Responder. When using SCVP, the Responder is responsible for performing the verification done by the client prior to submitting its request to the Responder when using OCSP. This provides benefits such as simplifying client implementations and allowing companies to centralize their trust and policy management. For the most part, client implementations and policy management are already simplified through the use of validation toolkits that encapsulate all the necessary client side functionality and are easy to integrate and configure. Additionally, client toolkits provide greater flexibility in terms of the various operational environments into which PKI's are being deployed.

CRL/CRL Deltas (VACRL protocol)

It is possible for the client to maintain its own store of certificate status information by obtaining either a cumulative CRL (used to initially establish the client's CRL store) or "CRL Deltas", updates to the CRL based on the age of the client's last obtained CRL. The transaction is conducted over HTTP or HTTP/SSL and the client uses a standard GET or POST operation to make its request. The Responder will either return the cumulative CRL signed by the CA or it will manufacture the appropriate CRL Deltas for that client and sign the response using its VA certificate (which the client will verify as per the discussion on trust models). The client uses the response it obtained from the Responder to build its own local CRL store. This allows a client application to validate certificates even when it does not have real-time access to the VA (as is necessary for OCSP) but addresses the scalability issue inherent in the client always having to obtain a cumulative CRL. This protocol is not mutually exclusive with OCSP and in fact is often used by enterprises with PKI in conjunction with OCSP as a backup alternative.

Tumbleweed Validation Products

The Valicert Validation Authority™ (VA) delivers a comprehensive, scalable, and reliable framework for validating digital certificates. It is comprised of the following components.

Enterprise Validation Authority (EVA) Server: Robust, fourth generation server that provides online validation support using OCSP, SCVP, CMP, and CSC standards. Integrates with major CA products and services including those from Baltimore Unicert, Entrust PKI, Microsoft CA, AOL/Netscape CMS and Sun CMS

VA Publisher: Obtains revocation data from a CA or a directory server supporting LDAP, LDAPS, HTTP, and HTTPS and publishes it to a Tumbleweed Valicert EVA

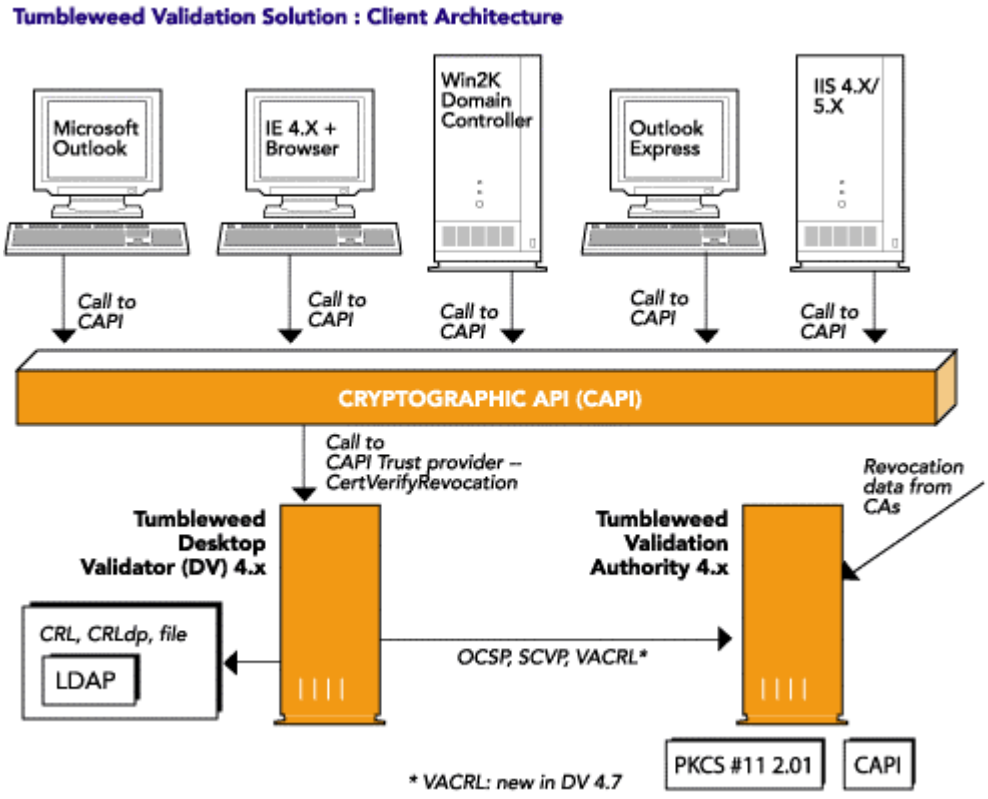

Figure 4:

Desktop Validator (DV): **[Figure 4]** a CAPI 2.0 compliant certificate revocation status checking provider for Windows 95/98/ME/XP/NT/2000/2003. DV provides revocation checking seamlessly for Microsoft applications (IE, Outlook, Outlook Express, IIS, Windows 2000 Domain Controllers and Windows 2000 server). DV provides certificate validation for secure e-mail, secure web access, VPN, and smart card login. With Microsoft Office XP, DV can perform validation for signed Word, Excel, and PowerPoint documents. DV can be deployed using Microsoft SMS for enterprise deployments. DV provides CA specific options that

allows for flexible rules to be defined for certificate status checking. DV provides robust fail-over support from multiple sources of revocation information. DV supports Microsoft and Sun/AOL proxy servers and proxy synchronization with IE for custom proxy. DV supports automatic configuration with EVA server.

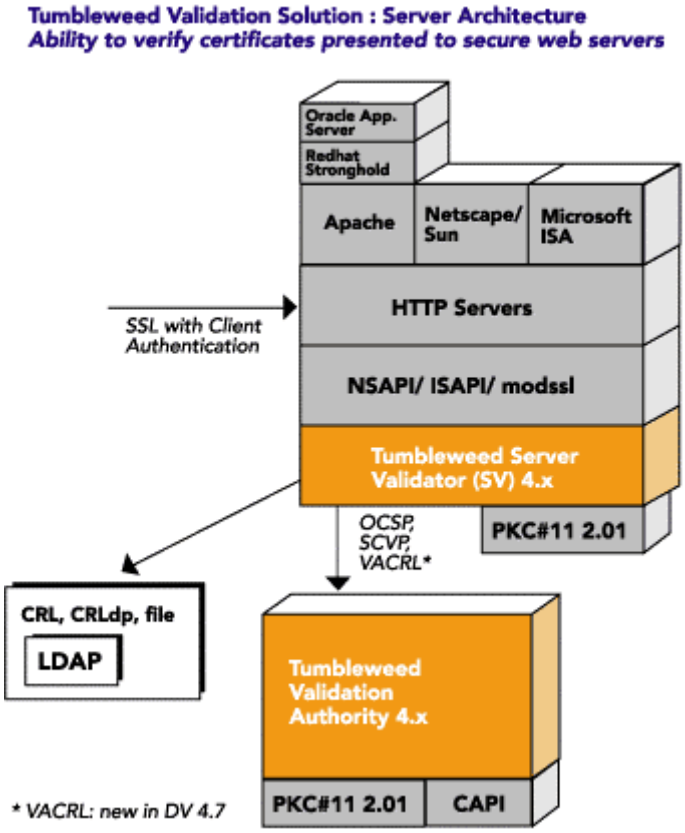

Figure 5:

Server Validator (SV): **[Figure 5]** provides certificate revocation status checking for secure web servers. Web servers requiring SSL and client authentication, can use SV to add revocation checking for certificates used to authenticate to web servers or web enabled applications. SV is supported on both Windows and UNIX. SV provides revocation checking within Microsoft ISA, Apache server (including Oracle Application Server and RedHat StrongHold variants), AOL/Netscape or Sun Web Server environments. SV supports automatic configuration with EVA server for easy server deployment.

Validator Toolkit: Easily adds digital certificate validation functionality to third-party and custom applications The toolkit includes a complete set of certificate validation functions, source code examples, reference manual. Available for C/C++, Java, UNIX and Windows environments. FIPS 140-1 / JITC version are also available.

Conclusion

The security objectives of a PKI cannot be met unless certificates are validated when presented. The introduction of a Validation Authority within a PKI enables a more robust, scalable architecture, and enables the entire infrastructure to be more secure. There are many different ways in which certificates can be validated, allowing organizations to choose the mechanism most suitable for their operational environment. A fourth generation product, the Tumbleweed/Valicert Validation Solution represents the most comprehensive solution for certificate validation and has been acknowledged as the superior solution by many different customers, including those with the most stringent security requirements.

Tumbleweed Communications Corp.
700 Saginaw Drive
Redwood City, CA 94063
www.tumbleweed.com

Tel: 650.216.2000
Fax: 650.216.2001
info@tumbleweed.com